Wrocław, 22.10.2012

Języki programowania – lista 4: proste problemy numeryczne, funkcje i wyjątki

1. W wykładzie 2, na slajdzie nr 49, przedstawiono program znajdujący medianę temperatur. Może on nie działać poprawnie, gdy zebrano parzystą liczbę temperatur.
   a) popraw program,
   b) uzupełnij go o liczenie średniej geometrycznej lub harmonicznej.

2. Napisz program będący bardzo prostym kalkulatorem. Niech obsługuje cztery podstawowe operacje: dodawanie, odejmowanie, mnożenie i dzielenie – na dwóch operandach. Program powinien prosić użytkownika o podanie trzech argumentów: dwóch wartości typu **double** oraz znaku reprezentującego rodzaj działania. Jeśli użytkownik wpisze liczby ***35.6*** i ***24.1*** oraz znak +, program powinien zwrócić komunikat ***„Suma liczb 35.6 i 24.1 wynosi 59.7”***.

3. Napisz program rozwiązujący równanie kwadratowe postaci ***a*x^2 + bx + c = 0***. Utwórz funkcję obliczającą i zwracającą pierwiastki tego równania po podaniu ***a, b*** oraz ***c***. Jest mały problem – jeśli ***b^2 – 4ac*** wyniesie mniej niż zero, to nie da się wykonać obliczeń. Zgłoś wtedy wyjątek. Niech ta funkcja będzie wywoływana przez funkcję główną programu, która powinna przechwycić wyjątek, jeśli zostanie zgłoszony. Jeśli program wykryje w ten sposób, że równanie nie ma rozwiązań w zbiorze liczb rzeczywistych – niech wydrukuje odpowiedni komunikat.

4. Napisz program znajdujący wszystkie liczby pierwsze z zakresu od **2** do **max** działający wg następującego pomysłu: Napisz funkcję sprawdzającą, czy daną liczbę można podzielić bez reszty przez mniejszą od niej liczbę pierwszą, której parametrem będą sprawdzana liczba oraz wektor liczb pierwszych. Następnie napisz pętlę działającą w zakresie od **2** do **max**, sprawdzającą czy kolejna liczba jest pierwsza i jeśli tak, dodającą ją do wektora liczb pierwszych. Wreszcie napisz pętlę wyświetlającą wszystkie odnalezione liczby pierwsze.

Dla ambitnych

5. Napisz program do gry w zgadywanie liczb. Użytkownik wymyśla sobie jakąś liczbę z zakresuvod **1** do **100**, a program próbuje odgadnąć, co to za liczba, zadając różne pytania (np. ***„Czy ta liczba jest mniejsza niż 50?”***). Program powinien być w stanie odgadnąć liczbę po zadaniu nie więcej niż siedmiu pytań.

6. Zaimplementuj prostą grę w zgadywanie i nazwie „byki i krowy”. Program ma wektor czterech cyfr, a zadaniem gracza jest odgadnąć te cyfry w kilku próbach. Jeśli do odgadnięcia jest liczba ***1234***, a gracz wpisze ***1359***, program powinien wydrukować odpowiedź „1 byk i 1 krowa”, ponieważ gracz odgadł jedną cyfrę na właściwej pozycji (byk) i drugą na złej pozycji (krowa). Gra toczy się, aż użytkownik odgadnie wszystkie cztery cyfry na właściwych pozycjach. Zamiast zapisywać wartość do odgadnięcia na stałe w programie, użyj generatora liczb pseudolosowych **rand()** z części biblioteki standardowej on nazwie **cstdlib.** Generator zwraca liczbę całkowitą z zakresu **0** do **RAND_MAX**. Aby zmienić zakres na **0..9** możesz użyć operacji modulo. Zauważ, że sekwencja kolejnych liczb pseudolosowych zawsze jest taka sama. Aby tego uniknąć, należy przed pierwszym losowanie wywołać funkcję ***srand(n)***, gdzie **n** jest ziarnem generatora, która powinno z każdym uruchomieniem programu być inne. Jak to osiągnąć? Można prosić użytkownika o podanie dowolnego **n**. Często jednak inicjuje się generator liczb losowych aktualnym czasem systemowym. Aby pobrać czas systemowy użyj funkcji **time(NULL).**

Powodzenia!
Witold Dyrka i zespół